1、数值拓展

1. 二进制和八进制

```
let b = 0b1010//二进制
let o = 0o777;//八进制
let d = 100;//十进制
let x = 0xff;//十六进制
console.log(b);
console.log(o);
console.log(d);
console.log(x);
```

2. Number.EPSILON：

它是 JavaScript 表示的最小精度，EPSILON 属性的值接近于 2.220446049250313080847263336181 6E-16

```
function equal(a, b) {
    if (Math.abs(a - b) < Number.EPSILON) {
        return true;
    } else {
        return false;
    }
}
console.log(0.1 + 0.2 === 0.3);//false
console.log(equal(0.1 + 0.2, 0.3));//true
```

3. Number.isFinite:检测一个数值是否为有限数

```
console.log(Number.isFinite(100)); //true
console.log(Number.isFinite(100 / 0)); //false
console.log(Number.isFinite(Infinity)); //false
console.log(Number.isFinite(-Infinity)); //false
```

4. Number.parseInt：将一个字符串转换为整数

```
console.log(Number.parseInt("123abc"));//123
```

5. Number.parseFloat：将一个字符串转换为浮点数

```
console.log(Number.parseFloat("3.1415926神奇"));//3.1415926
```

6. Number.isInteger：判断一个数是否为整数

```
console.log(Number.isInteger(5));//true
console.log(Number.isInteger(2.5));//false
```

7. Math.trunc：将数字的小数部分抹掉

```
console.log(Math.trunc(3.5));// 3
```

8. Math.sign：判断一个数到底为正数、负数、还是零

```
console.log(Math.sign(100));//1
console.log(Math.sign(0));//0
console.log(Math.sign(-20000));//-1
```

2、JSON实现深拷贝

JSON.parse(JSON.stringify(obj))：可实现多维对象的深拷贝，但会忽略 undefined 、 任意的函数 、Symbol 值

```
var obj1 = {
    name: "张三",
    age: 21,
    birthday: {
        year: 2000,
        month: 12,
        day: 5
    },
    speak: function () {
        console.log("我是" + this.name);
    }
};

var obj2 = JSON.parse(JSON.stringify(obj1));
```

```
15
16 // 当修改obj2的属性和方法的时候，obj1相应的属性和方法不会改变
17 obj2.name = "李四";
18 console.log(obj1);
19 console.log(obj2);
```

3、对象方法拓展

```
1  //1. Object.is 判断两个值是否完全相等
2   console.log(Object.is(120, 120)); // ===
3   console.log(Object.is(NaN, NaN)); // ===
4   console.log(NaN === NaN); // ===
5
6  //2. Object.assign 对象的合并
7  const config1 = {
8      host: 'localhost',
9      port: 3306,
10     name: 'root',
11     pass: 'root',
12     test: 'test'
13 };
14 const config2 = {
15     host: 'http://atguigu.com',
16     port: 33060,
17     name: 'atguigu.com',
18     pass: 'iloveyou',
19     test2: 'test2'
20 };
21 console.log(Object.assign(config1, config2));
22
23 //3. Object.setPrototypeOf 设置原型对象  Object.getPrototypeof
24 const school = {
25       name: '尚硅谷';
26  }
27 const cities = {
28       xiaoqu: ['北京', '上海', '深圳'];
29 }
30 Object.setPrototypeOf(school, cities);
31    console.log(Object.getPrototypeOf(school));
```

```
32    console.log(school);
```